Exhibit A

Intelligence to Right-price IP Services

xacct

To: Limor, Anil, Willie, Eran, Eric
Subject: **IP Accounting Under Attack – Let's Patent it – NOW!**
CC: Yuval, Ori, Gil H
From: Tal Givoly
Date: August 17, 1999

Overview

This memo is to prepare for the meeting we have on Monday. I realize this may be a bit late, but I hope you have time, at least, to go over this once during the weekend so we can discuss it further when we meet.

Agenda

Discuss patents already in the pipe:

- IP Mediation
- Session Reconstruction

Questions:

- What is the actual filing date?
- When will they be granted?
- How's it going with the filing in England? When is this done?

List of potential patents (more details on each, below):

- IP Accounting Under Attack
- Distributed session reconstruction (can be an extension to Session Reconstruction, above)
- A library providing provisioning of accounting element, aggregation, filtering, at the network/service element
- Distributed drill-down into distributed databases

Questions:

- What's worth moving forward on?
- What's the next steps? Where should we patent? What can be done to accelerate?

IP Accounting Under Attack

This section presents an idea, which if patented, can potentially prevent any other mediation/billing solution from entering the IP billing mediation space.

The Problem

We discovered at several customers (Navy, Harvard, Xpert) that when a network is under a hostile attack (an attempt to break in), a huge surge occurs in the amount of accounting information (not network traffic) that is generated by devices.

For instance, if a computer on the network attempts a syn or fin attack on a network, it will scan all ports. There are 65,536 ports to scan, and all this takes place over a very short period of time, typically several seconds. The amount of network traffic that this represents is usually negligible (as the packets are of minimal size), but the amount of accounting data created is large. For instance, it would create 65,536 log entries in a firewall log, or 131,072 NetFlow flows, for each host that it attempts to attack. If a ping attack is used, then all the IPs are scanned in a similar fashion.

The Solution

The idea is that since this represents a serious problem for the mediation or billing system to cope with, it should be classified on ingress point to the mediation solution and a single record characterizing the attack should be produced. For instance:

A <type of attack> attack from <host> started at <start> lasted <duration> seconds and scanned through <IP range> and <Port range>.

This record could easily aggregate hundreds of thousands of NetFlow flows or records from a firewall log into a single record. Without this, the mediation/billing could easily overflow causing loss of data.

Compared to Telephony

This problem does not exist, to this degree, in the telephony world. There are cases in which there is a surge in the use of telephones, e.g. public gatherings, but it isn't in the order of magnitude that can be generated by an automated mechanism, such as a computer program. In telephony, this feature is called "mass call detection" and has nothing to do with automated, methodical, attack.

So it makes perfect sense to patent this technology and incorporate into our product in the ISMs that this may affect (NetFlow, FWs, RMON, etc.). Every mediation/billing vendor will essentially NEED to implement a mechanism similar to this. The fact that we will patent many ways to detect this and protect the mediation/billing will allow us to prevent them from infringing on our patents and make it impossible to use competing solutions.

A Side Effect

A side-effect advantage is that this also creates an attack-detection/warning mechanism built into those ISMs that employ this technology. The ISMs could create some trap sent to an NMS to handle the attack.

The Technical Detail

Below is a list of techniques that we will patent as being used to identify attacks from accounting data. More can be added, of course, to make this as complete as possible and to prevent competition from suggesting a viable solution without infringing on our patent.

- Detect port scan, either up or down. Be careful not to trigger a false alarm on OS-randomly-assigned source ports.
- Detect IP scan.
- Detect a surge in accounting traffic rate in ingress. In this case, count and discard records above a threshold. The attack is by the mere suggestion that this is not reflective of normal usage patterns.

Conclusion

We need to patent this technology today, implement it tomorrow and announce it the next day. This could be a very important competitive advantage in light of the emerging competitive scenarios we're about to face.

Distributed Session Reconstruction

The Problem

When performing session reconstruction in order to identify the attributes of the higher layers of the protocol stack (usually, layer 7 – application layer), it is imperative to capture all the packets in a given session. The control channels (containing the various negotiations and attributes) are particularly sensitive to loss in data captured. For example, if the negotiation specifying on which a port an FTP will transfer the data is not captured and analyzed, the bulk of the FTP transfer will not be captured and reconstructed. Usage data recorded will be inaccurate or, at least, unidentified to the protocol.

This type of problem occurs more frequently as networks become switched, and interconnect between POPs (Points of Present) of a service provider and between service providers become more meshed and redundant. The redundancy is used during normal use as load-balancing. Two or more links carry, potentially, the same traffic. The decision regarding which physical link to traverse depends on the actual load. This type of dynamic load balancing causes a single session, perhaps, to be present on different physical or logical communication channels. Performing session reconstruction by promiscuous reading of packets from one channel may miss pertinent information, as mentioned above.

The Solution

A single computer process that would capture all the data would resolve this problem. As this is not feasible due to the distributed nature of the If the amount of data wa

Conclusion